

CONTROL #1

Python para el análisis de datos

Fecha publicación: MARTES 04 de JUNIO de 2024

Fecha entrega: **JUEVES 13 DE JUNIO DE 2024**

Entregas posteriores a la fecha de entrega tienen un descuento 10 décimas por día de atraso.

 **Importante.** Lea detenidamente el enunciado. Le solicitamos encarecidamente que lea las instrucciones antes de comenzar a resolver los problemas. La fecha de entrega es el JUEVES 13 de JUNIO hasta las 23:59hrs.

INSTRUCCIONES

1. Resuelva **únicamente tres problemas de esta guía**. Cada problema debe resolverlo en un archivo independiente, y **el nombre del archivo debe corresponder con el número del problema**. Por ejemplo si resuelve los problemas: {1, 3 y 5}, **debe escribir su solución en tres archivos cuyos nombres sean: {problema_1.py, problema_3.py, problema_5.py}**. Una vez resuelto, debe subir **cada archivo** a la página de webcursos con el link: **"CONTROL_ENTREGA_1"**
2. Este control tiene un puntaje máximo de 60 puntos. Si resuelve más de tres problemas, solo serán considerados los primeros tres en orden creciente.
3. Este control debe ser resuelto en forma individual.
4. Asegúrese de leer todos los pasos anteriores. Recuerde que debe resolver esta tarea en Python (solo en Python)
5. Revise nuevamente desde el primer punto de este instructivo si algo no quedó claro.

PROBLEMA 1 (20 puntos)

Una forma matemática para determinar el valor de la función seno es a través de la expansión de una serie de Taylor. Dicha serie define la función seno de la siguiente forma

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \cdot \frac{x^{2n+1}}{(2n+1)!}, \text{ por lo tanto, } \sin x = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{(2n+1)!}$$

Implemente un programa en Python que:

- a) **Calcule** el seno de x (*sin x*, donde x debe ir definido en radianes). Asegúrese que el programa verifique que se ingrese un número entero (ver **Hint**). Puede emplear una biblioteca matemática (`math`) para calcular el factorial de un número.

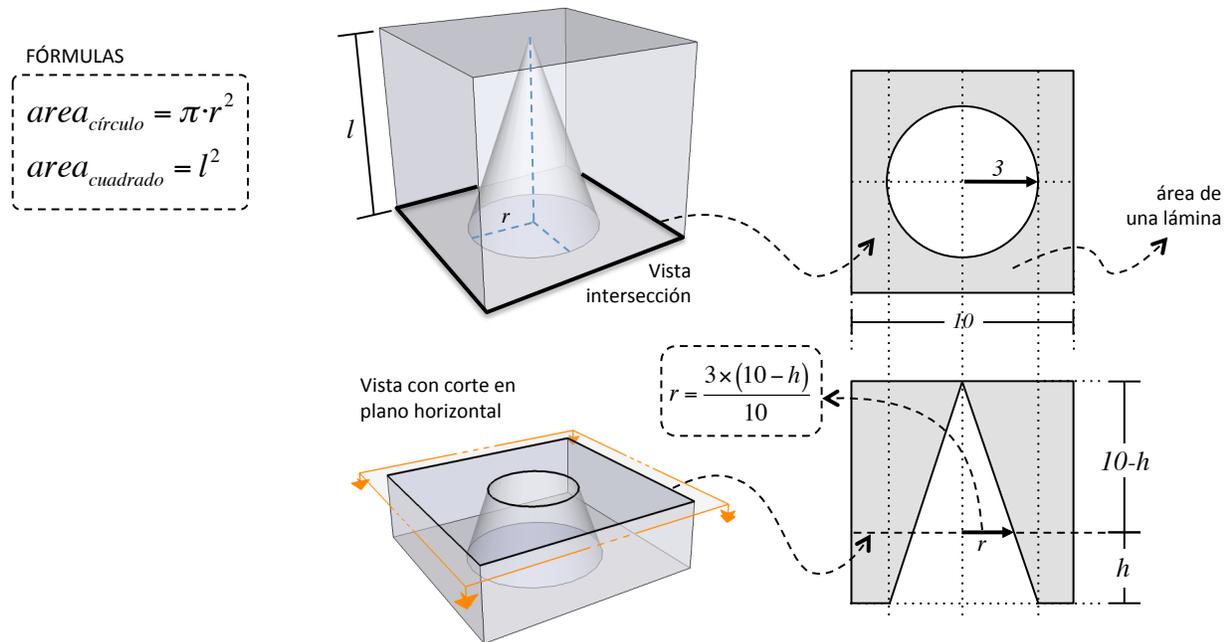
Hint: `variable.isdigit()`

IMPORTANTE. Su programa debe realizar la cantidad de iteraciones que el usuario defina, es decir, debe solicitar por teclado el valor de la variable x y la variable n . El valor x que usted debe ingresar debe estar en grados sexagesimales, es decir, valores entre 0 a 360 (ambos inclusive), sin embargo, debe cambiar dicho valor a radianes dentro de su programa.

PROBLEMA 2 (20 puntos)

Un profesor de cálculo ha solicitado aproximar el volumen generado por la **diferencia entre un cubo de lado $L = 10$ y un cono de radio $r = 3$ de altura $h = L$** (es decir, la misma altura que el cubo).

El profesor explica que dividiendo el problema en infinitos cálculos de áreas de una lámina (**de espesor 0.0001**), puede calcular el volumen de la figura sumando las láminas con diferente área. Para ayudarlos, el profesor les hace ver que a medida que "subimos" en la altura del cono (variable h), el radio r de la circunferencia decrece en función de esta altura h , según presenta la figura a continuación,



Hacer esto manualmente sería muy complicado, por lo que le pide desarrollar un programa en Python de forma que podamos variar la altura (variable h) con un valor muy pequeño (épsilon), y luego calcular el valor del radio r para cada altura h , y finalmente calcular el volumen como la suma de todas las láminas.

Implemente un programa en Python que:

- Incremente** la altura h para estimar el volumen generado por la **diferencia entre el cubo y el cono según** las condiciones antes descritas.
- Muestre** el resultado del volumen final por pantalla. Nota: considere que el valor de altura aumenta en $\epsilon = 0.0001$.

Recuerde que el área del círculo se calcula como $\pi * r^2$, donde r es el radio, y el área del cuadrado está dada por l^2 , donde l es la longitud del lado. El número Pi está incorporado en la biblioteca de math.

PROBLEMA 3 (20 puntos)

Dada su experiencia en el análisis de secuencias de ADN, le fue solicitado encontrar un patrón específico dentro del genoma humano. Como usted sabe, el código genético está compuesto por cuatro letras {A,T,C,G}. Una codificación genética puede ser, por ejemplo: AATTGTAGTA. Su tarea consistirá en **buscar en una secuencia de ADN ingresada por teclado el patrón A?T?**, donde “?” significa que puede ser cualquiera de las cuatro letras {A,T,C,G} (ver ejemplo en la Figura 1).

Implemente un programa en Python que:

- Solicite** al usuario una cadena de texto de no más de 100 caracteres.
- Reciba la cadena de texto**
- Muestre por pantalla** el número de veces que aparece el patrón **A?T?** dentro de la secuencia ingresada (ver ejemplo Fig.1).

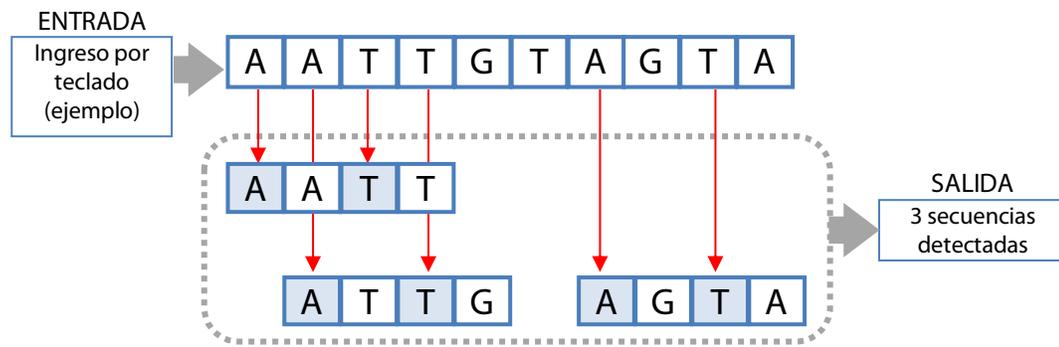


Figura 1. Ejemplo de secuencia de ADN

PROBLEMA 4 (20 puntos)

Un acertijo que se les plantea a los niños cuando están aprendiendo lógica matemática es el de la "rana en el pozo". En este acertijo los niños deben responder cuántos días tarda una rana en salir de un pozo de cierta profundidad, dado que de día avanza una determinada cantidad de metros y por la noche retrocede una cierta cantidad de metros.

Por ejemplo, **supongamos que el pozo tiene 31 metros de profundidad**, la rana avanza 5 metros durante el día y desciende 2 metros por la noche. Supongamos además que la rana inicia sus intentos de día. En tal caso, la rana demorará 10 días en salir. La Tabla 1 muestra el detalle del avance diario de la rana con estos datos de entrada (profundidad = 31, avance=5, retroceso=2).

Tabla 1. Ejemplo de avance y retroceso

Día	Avanza	Retrocede	Situación Final
1	5	2	3
2	5	2	6
3	5	2	9
4	5	2	12
5	5	2	15
6	5	2	18
7	5	2	21
8	5	2	24
9	5	2	27
10	5		Sale del pozo

Implementa un algoritmo que permita **resolver el problema de forma genérica**. Es decir, determina cuántos días demora la rana en salir de un pozo **para cualquier combinación de valores para la profundidad, el avance y el retroceso (en metros)**. Suponga que la rana siempre inicia sus esfuerzos de día. Escriba un algoritmo en Python que implemente dicha solución.

Implemente un programa en Python que:

- Solicite** al usuario ingresar la profundidad en metros del pozo
- Solicite** al usuario ingresar la cantidad de metros que la rana avanza de día
- Solicite** al usuario ingresar la cantidad de metros que la rana cae de noche
- Calcule** el número de días que demora la rana en salir y **desplegar** esta información por pantalla

La Figura 1 muestra un ejemplo de cómo debe ser la interacción del algoritmo con el usuario.

Figura 1. Ejemplo de ejecución de este algoritmo (parte A)

```
Ingrese la altura en metros del pozo (en metros): 31
Ingrese la cantidad de metros que avanza de día (en metros): 5
Ingrese la cantidad de metros que cae de noche (en metros): 2

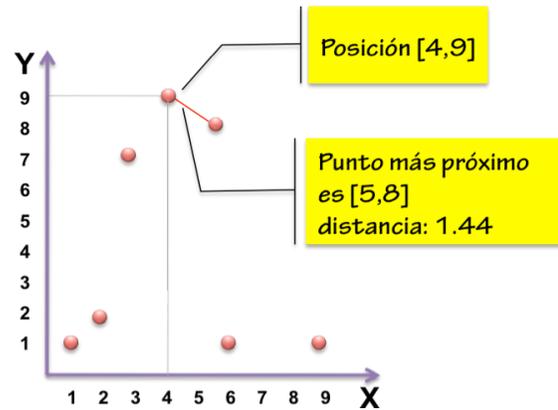
La rana se demora 10 días en salir
```

PROBLEMA 5 (20 puntos)

Suponga que tiene dos vectores de igual tamaño de largo 10 que representan puntos en el espacio cartesiano X-Y. Implemente un programa en Python que **por cada coordenada** determine la coordenada más cercana. Indique dicha distancia. (**Hint:** Ocupe ciclos dobles)

Veamos el siguiente ejemplo para los siguientes datos de entrada.:

```
X = [2, 3, 6, 6, 4, 1, 9]
Y = [1, 7, 1, 8, 9, 1, 1]
```



Implemente un programa en Python que:

- Solicite 10 valores al usuario para cada variable (10 para la variable X y 10 para la variable Y)
- Muestre la distancia más cercana para cada coordenada ingresada.

PROBLEMA 6 (20 puntos)

Una compañía distribuidora de aceite para vehículos motorizados ha decidido mecanizar el control de las ventas en uno de sus puntos de distribución. La compañía comercializa tres tipos de aceites: aceite tipo 1 (de primera), cuyo valor por litro es de \$ 5.000, aceite tipo 2 (de segunda) con precio de \$ 4.000 por litro y aceite tipo 3 (de tercera) con costo de \$ 3.000 por litro.

Durante el fin de semana pasado se registró un incremento en las ventas llegando a casi 400 clientes, lo que dificulta la administración y control de estos insumos. Como especialista en el área de ciencias de datos, se solicita implementar una solución computacional -en lenguaje Python- que permita registrar las ventas que realiza esta compañía por cada tipo de aceite.

Implemente un programa en Python que:

(a) **Lea** desde teclado para cada tipo de aceite el total de litros vendidos durante cada día del mes. El programa debe generar los **arreglos T1, T2 y T3** de largo 31 **con valores correspondiente a los litros de aceite vendidos por día**. Ejemplo:

	T1	T2	T3
31	314	123	23
	123	34	13
	992	23	114
	992	11	445
	35	345	532

Usted define una forma de ingresar esta información por teclado, y ésta debe ser indicada al usuario por pantalla.

(b) **Calcule** e imprima el total facturado (litros vendidos X precio por litro) **por cada tipo de aceite** y retorne el **Total facturado de todos los aceites** [T1+T2+T3]

(c) **Retorne** el tipo de aceite con mayor cantidad de litros vendidos: {T1, T2, o, T3}

(d) **Retorne el tipo de aceite con mayor ganancia.** Hint: litros vendidos X precio por litro

PROBLEMA 7 (20 puntos)

Los profesores de cálculo quieren demostrar a los alumnos que una forma de calcular la raíz cúbica de un número es iterar la siguiente fórmula

$$X = (2 * X^3 + \text{num}) / (3 * X^2)$$

hasta que el valor absoluto de $X_{\text{actual}} - X_{\text{anterior}}$ sea menor que un cierto valor dado que llamaremos ϵ .

Por ejemplo, para calcular la raíz cúbica de 8 con un valor de ϵ de 0.001, al algoritmo repite el cálculo de la fórmula de la siguiente forma:

$X = 1.000$	# X inicial
$X = (2 * 1.000^3 + 8) / (3 * 1.000^2) = 3.333$	# $3.333 - 1.000 = 2.333$, que es mayor a ϵ
$X = (2 * 3.333^3 + 8) / (3 * 3.333^2) = 2.462$	# $3.333 - 2.462 = 0.871$, que es mayor a ϵ
$X = (2 * 2.462^3 + 8) / (3 * 2.462^2) = 2.081$	# $2.462 - 2.081 = 0.381$, que es mayor a ϵ
$X = (2 * 2.081^3 + 8) / (3 * 2.081^2) = 2.003$	# $2.081 - 2.003 = 0.078$, que es mayor a ϵ
$X = (2 * 2.003^3 + 8) / (3 * 2.003^2) = 2.000$	# $2.003 - 2.000 = 0.003$, que es mayor a ϵ
$X = (2 * 2.000^3 + 8) / (3 * 2.000^2) = 2.000$	# $2.000 - 2.000 = 0$, que es menor a ϵ , entonces termina la ejecución.

Los profesores de cálculo no son buenos programadores, así que nos han solicitado a los profesores de Python que lo hagamos... y nosotros se lo pedimos a Ustedes (que ironía, ¿no?).

Implemente un programa en Python que:

- Lea por teclado un número **num**
- Retorne el valor aproximado de la raíz cúbica de este número utilizando la fórmula anterior. Considere un valor de **ϵ igual a 0.001** y un **valor inicial de X_{actual} igual a 1** y un **valor inicial de X_{anterior} igual a 0**.

Recuerde emplear la biblioteca **math** ya que esta contiene la función **pow(y,z)** que calcula **y elevado z**, y la función **fabs(num)** que retorna el **valor absoluto de num**

PROBLEMA 8 (20 puntos)

La gran multitienda `LeroMart` tiene una oferta lleve 3 y pague 2 la cual consiste en poder comprar 3 elementos cuales quiera y solo pagar 2 (la tienda no es tonta por lo que cobrará los 2 elementos más caros), la promoción no aplica si lleva más de 3 elementos. Alberto tiene una lista de N elementos que desea comprar (asuma que todos los elementos existen en `LeroMart`). Alberto es muy inteligente por lo que decide entrar las veces que sean necesarias para pagar menos. Por ejemplo, si entra 2 veces comprando 3 elementos cada vez, pagará sólo 4 elementos en lugar de pagar los 6.

Figura 1. Ejemplo de ejecución de este algoritmo

```
cuantos productos? 7
2000 4000 1000 2000 4000 3000 2000

El valor mínimo que podría pagar es: 13000
```

Implemente un programa en Python que:

- Lea** por teclado el número de productos que el usuario desea comprar
- Lea** por teclado los precios de cada uno de los productos a comprar
- Retorne** como resultado el valor mínimo que podría pagar por dichos productos.

PROBLEMA 9 (20 puntos)

Usted decide finalmente que ahora es la mejor época para invertir su dinero. Solicita a distintos bancos tasas de interés y elige la mejor de ellas. Suponga que la tasa de interés del mejor banco es del $i\%$, y usted deposita mensualmente x pesos en p meses.

Ejemplo: Interés $i=5\%$, Ahorro mensual $x= 100$, $p=3$ meses

Mes (día 1)	Ahorro	Total (al final del mes)
1	$100 + 0 + (100 + 0) \times 0.05$	\$105
2	$100 + 105 + (100 + 105) \times 0.05$	\$215.25
3	$100 + 215.25 + (100 + 215.25) \times 0.05$	\$331.0125

En el ejemplo anterior, según dichos valores el programa debe retornar el número \$331.0125

Implemente un programa en Python que:

- Determine el monto final del ahorro ingresando los tres parámetros mencionados anteriormente (i , x , p)
- Retorne el resultado del cálculo.

PROBLEMA 10 (20 puntos)

Usted estaba visitando una antigua ciudad Árabe cuando inesperadamente se encuentra con la lámpara de Aladino en el suelo. Al frotarla, aparece el "mago de Aladino" el cual le dice pide un deseo y te será otorgado. Luego de reflexionar un momento decide que la riqueza no es lo más importante en su vida, por lo cual, le pide a Aladino el siguiente deseo: "*Deseo que cada plato de comida que yo aporte a una persona necesitada, tu duplicas el aporte total*", realice un programa que.

Ejemplo:

Ejemplo: Tabla de desarrollo de aporte de platos de Aladino

Día	Plato de comida aportado por mi	Acumulado	Aporte de Aladino en platos
1	1	1	$2*1=2$
2	1	2	$2*3=6$
3	1	6	$2*7=14$
4	1	14	$2*15=30$
5	1	30	$2*31=62$
??	1	<1.000.000.000	>1.000.000.000

Implemente un programa en Python que:

- Ingrese por teclado el número de personas que se encuentra bajo la línea de pobreza (ej. Mil millones)
- Determine el número de días tomaría su deseo para que todos ellos sean alimentados **para un número de personas declarado en el paso anterior**

PROBLEMA 11 (20 puntos)

Ha salido al mercado un nuevo juego conocido como "aplantar la rana". El juego consiste en atrapar la rana antes que ella se mueva de posición. Para ello usted debe llenar una matriz de 3x3 con letras X, pero siempre debe asegurar que en alguna **posición aparezca la letra R en la matriz**. Para aplantar la rana, el usuario debe dirigirse donde se encuentre la rana y así alcanzarla.

Los movimientos permitidos son: **letra E para subir, letra X para bajar, letra D para ir a la derecha y letra S para ir a la izquierda**. Cada vez que el usuario presiona una tecla, la rana salta de posición a otro lugar dentro de la matriz. **El jugador siempre comienza en la posición 1,1 y con letra J. La rana puede saltar en cualquier posición dentro del tablero**. Cuando la rana desaparece, significa que el jugador aplastó la rana y el juego finaliza. El jugador que haga menos movimientos gana la partida.

Ejemplo de juego:

suponga que la matriz se llena con las siguientes letras al inicio del juego (**inicio**). La siguiente secuencia podría ser los movimientos del jugador y la rana.

X	R	X
X	J	X
X	X	X

X	J	X
R	X	X
X	X	X

J	X	X
X	X	X
R	X	X

X	X	X
J	X	X
X	X	X

Inicio:

Jugador presiona la tecla E (sube una posición)

1er movimiento

Jugador presiona la tecla S (se mueve a la izquierda)

2do movimiento

Jugador presiona la tecla X (se mueve hacia abajo)

3er movimiento

La rana desaparece del tablero. Ganaste el juego 😊

Implemente un programa en Python que:

- Genere una matriz con letras al inicio del juego
- Permita al jugador moverse dentro del tablero con las teclas E, X, F, S.
- Determine el momento en que el jugador aplasta a la rana.

PROBLEMA 12

Seguramente has visto un *display de siete segmentos*.

Es un dispositivo (a veces electrónico, a veces mecánico) diseñado para presentar un dígito decimal utilizando un subconjunto de siete segmentos. Si aún no sabes lo qué es, consulta la siguiente liga en Wikipedia: https://en.wikipedia.org/wiki/Seven-segment_display.

Tu tarea es escribir **un programa que puede simular el funcionamiento de un display de siete segmentos**, aunque vas a usar LEDs individuales en lugar de segmentos.

Cada dígito es construido con 13 LEDs (algunos iluminados, otros apagados, por supuesto), así es como lo imaginamos:

Nota: el número 8 muestra todas las luces LED encendidas.

Tu código debe *mostrar* cualquier número entero no negativo ingresado por el usuario. Consejo: puede ser muy útil usar una lista que contenga patrones de los diez dígitos

Datos de prueba

Entrada de muestra:

123

Salida de muestra:

```
# ### ###
#  #  #
# ### ###
# #   #
# ### ###
```

Entrada de muestra:

9081726354

Salida de muestra:

```
### ### ### # ### ### ### ### ### # #
# # # # # # # # # # # # # #
### # # ### # # ### ### ### ### ###
# # # # # # # # # # # # #
### ### ### # # ### ### ### ### #
```